IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC D/B/A/ IXO,	§	
	§	
Plaintiff,	§	
v.	§	No. 11 Civ. 6696 (KBF)
	§	No. 11 Civ. 6701 (KBF)
	§	No. 11 Civ. 6704 (KBF)
MORGAN STANLEY, ET AL.,	§	
	§	
	§	
Defendants.	§	ECF Case

----------X

REALTIME DATA, LLC D/B/A/ IXO,	§	
	§	
Plaintiff,	§	
v.	§	No. 11 Civ. 6697 (KBF)
	§	No. 11 Civ. 6699 (KBF)
	§	No. 11 Civ. 6702 (KBF)
CME GROUP INC., ET AL.,	§	
	§	
	§	
Defendants.	§	ECF Case

----------X

REALTIME DATA, LLC D/B/A/ IXO,	§	
	§	
Plaintiff,	§	
v.	§	No. 11 Civ. 6698 (KBF)
	§	No. 11 Civ. 6700 (KBF)
	§	No. 11 Civ. 6703 (KBF)
THOMSON REUTERS, ET AL.,	§	
	§	
	§	
Defendants.	§	ECF Case

----------X

DECLARATION OF MICHAEL IAN SHAMOS, PH.D., J.D., IN SUPPORT OF PLAINTIFF REALTIME'S CLAIM CONSTRUCTION BRIEF

I, Michael Ian Shamos, hereby declare under penalty of perjury that the facts set forth herein are true and correct to the best of my knowledge and belief, and, if called as a witness, I can and will testify at trial as to the matters discussed in this Declaration.

1. I have been retained as an expert in this case by counsel for Plaintiff Realtime Data, LLC d/b/a/ IXO ("Realtime") to provide my opinions in connection with U.S. Patent Nos. 7,417,568 (the "'568 Patent"), 7,777,651 (the "'651 Patent"), and 7,714,747 (the "'747 Patent") (collectively the "Patents").

A. QUALIFICATIONS

2. My name is Michael I. Shamos. I hold the title of Distinguished Career Professor in the School of Computer Science at Carnegie Mellon University ("Carnegie Mellon") in Pittsburgh, Pennsylvania. I was a founder and Co-Director of the Institute for eCommerce at Carnegie Mellon and I now direct a graduate degree program in eBusiness Technology.

3. I teach graduate courses at Carnegie Mellon in Electronic Commerce, including eCommerce Technology, Electronic Payment Systems, Electronic Voting and eCommerce Law and Regulation and have done so since 1999. Since 2001, I have taught an annual course in Electronic Payment Systems as Visiting Professor at the University of Hong Kong. Since 2007, I have taught an annual course in Law of Computer Technology at Carnegie Mellon.

4. From 1979 to 1987, I was the founder and president of two computer software development companies in Pittsburgh, Pennsylvania: Unilogic, Ltd. and Lexeme Corporation.

5. I am an attorney admitted to practice in Pennsylvania and have been admitted to the Bar of the U.S. Patent and Trademark Office since 1981. I have not been asked to offer any opinions on patent law in this action.

6. I have previously testified in a number of cases concerning electronic auctions and electronic payment system.

7. I have been provided with the Patents, their prosecution histories, the reexamination histories, the parties' agreed claim constructions, the parties' proposed claim constructions for terms whose meanings are disputed, Realtime's infringement contentions and drafts of Realtime's claim construction brief.

B. LEVEL OF SKILL IN THE ART

8. The specification of the '651 Patent states:

> The present invention relates generally to systems and method for providing data transmission, and in particular, to systems and method for providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data and news feeds, advertisements, and all other forms or information over a communication channel using data compression and decompression to provide data broadcast feeds, bi–directional data transfers, and all other forms of communication with or without security and effectively increase the bandwidth of the communication channel and/or reduce the latency of data transmission.

'651 Patent at 1:19-30.

9. Not all of the recited technologies relate to the claims asserted in this action (the "Asserted Claims"). Therefore, based on the specifications of the Patents and the Asserted Claims, one of ordinary skill in the art would have to be familiar with methods for providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data and news feeds, advertisements, and all other forms or information over a communication channel using data compression and decompression to provide data broadcast feeds and bi–directional data transfers.

10. In consequence, it is my opinion that one of ordinary skill in the art would have had an undergraduate degree in computer science or electrical engineering, or equivalent work

experience, and in addition would be familiar with data compression and decompression methods, bi-directional data transfers and the characteristics of financial data streams.

C. CLAIM CONSTRUCTION

11. The parties disagree as to the meaning of certain claim terms. Below I explain the basis for my opinion that Realtime's proposed constructions are appropriate. I personally participated in the development of Realtime's constructions.

content dependent data decompression/content independent data decompression

12. These terms are used in the '747 and '651 Patents. Defendants assert that these terms are indefinite under 35 U.S.C. §112 even though they are well-known terms of art. Furthermore, Defendants do not contend that the parallel terms "content dependent data compression" and "content independent data decompression" are indefinite. It is clear to one of skill in the art that the term "content dependent data decompression" means "a decompression technique to reverse the compression of data achieved through content dependent data compression." This follows from the clear meaning of the English prefix "de," which means reversal.

13. Likewise, "content independent data decompression" means "a decompression technique to reverse the compression of data achieved through content independent data compression." I see no basis for Defendants' contention that the terms are indefinite. If one compresses, then one must decompress to recover the original data, and the decompression algorithm is the reverse of the compression algorithm.

data block type/data field type

14. These terms are used in the '568, '747 and '651 Patents. For "data block type," Realtime proposes "an attribute or characteristic of the data block." For "data field type, "

Realtime proposes "an attribute or characteristic of the data field." Defendants, on the other hand, propose the extremely narrow construction, "categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned integers, pointers, or other data type."

15. There is no basis for Defendants' restrictive construction. The concept of a "data block type" or "data field type" in the Patents is that different compression algorithms may be appropriate based on the nature of the data block or data field being compressed. This is expressed in the '568 Patent as follows:

> Because a multitude of different data types may be present within a given input data stream, or data block, to it is often difficult and/or impractical to predict the level of compression that will be achieved by any one encoding technique. Indeed, rather than having to first identify the different data types (e.g., ASCII, image data, multimedia data, signed and unsigned integers, pointers, etc.) comprising an input data stream and selecting a data encoding technique that yields the highest compression ratio for each of the identified data types, content-independent data compression advantageously applies the input data stream to each of a plurality of different encoders to, in effect, generate a plurality of encoded data streams. The plurality of encoders are preferably selected based on their ability to effectively encode different types of input data.

'568 Patent at 14:1-15.

16. Therefore, there is some attribute or characteristic of the data block or data field that makes it amenable to a particular method of compression. That is what is meant by "data block type" and "data field type."

17. Defendants have seized on the exemplary language of the above citation, "e.g., ASCII, image data, multimedia data, signed and unsigned integers, pointers, etc." and have attempted to turn that exemplary language into a claim limitation. To do so is to ignore the fact that the inventors expressly qualified that language as only representing an example. This is

impermissible as (1) reading a feature of a disclosed embodiment into a claim limitation and (2) treating exemplary language as limiting.

18. Moreover, Defendants appear to take the position that "data block type" means the "data type" of a "data block." This interpretation makes no sense because a data block can contain multiple items of information, each having different characteristics and each having a different data type. Defendants also appear to take the position that "data field type" means the "data type" of a "data field." This is incorrect and is contrary to the teaching of the Patents. Consider two examples, A and B. In both examples A and B, the data field has type 16-bit Integer. In example A, the values of the Integers range from 0 to 65,535 (the maximum range of a 16-bit integer). In example B, suppose it is known a priori that all the values are either 77 or 18,909. Clearly, the field in example B can be represented as a single bit. The value 77 can be represented by 0 and 18,909 can be represented by 1. This achieves a compression ratio of 16 because a 16-bit field in example B can always be represented as a 1-bit field. Such compression is not possible in example A, yet both data field A and data field B are 16-bit Integer. It is clear, therefore, that the method of compression must depend on more than just the "data type" of the field. The same is true, yet more profoundly, for a data block because the components of a data block do not even necessarily have the same data type.

19. Thus there is no sound basis for Defendants' construction.

determinate point

20. This term is used in the '651 Patent only. It also appears only in the claims. "Determinate point" is not a term of art and therefore is to be given its plain and ordinary meaning to one of skill in the art reading the specification. It means "a point (place) capable of being determined." To avoid simply repeating the words of the term, Realtime's construction

captures its meaning: "a discernable location." Defendants, on the other hand, have developed a definition of their own that finds no support in the specification: "an identifiable sequence of one or more bytes in the data stream." None of the following phrases appear in the specification "identifiable sequence," "one or more bytes," or "bytes of the data stream."

21. The context of '651 claim 15, e.g., which includes this term is "The system of claim 13, wherein the memory resets the adaptive table at a determinate point of the data packet." This claim means what it says, namely that at some point in the data packet, capable of determination, the memory resets the adaptive table. (Essentially, the prior adaptive table is discarded and a new one is built because the former adaptive table is no longer relevant to compressing the data that follows.) There is no teaching that such a point must be identified by a sequence of bytes, though it might be so identified. Yet there are other ways of identifying the determinate point. It might always occur, for example, after the 50th byte, whether or not there is any particular sequence of bytes in the data stream. I believe Defendants have confused "determinate point" with "synchronization point."

global state machines

22. This term is used in the '651 Patent. Realtime's construction is "hardware, microinstruction code, or application program whose state transitions are based on prior knowledge of at least some aspects of the content or structure of the financial data stream." Defendants' construction is "hardware, microinstruction code, or application program whose state transitions are based on prior knowledge of at least the data stream packet structure."

23. It may be useful here to explain what one of skill in the art would understand by a "state machine," a "global state machine" and a "local state machine." The notion of a "state machine" has been fundamental in computer science since at least the 1950s. One of the

simplest computing models known is that of the "finite state machine," which is a theoretical construct in which a system at any time can be in only one of a finite number of "states" or conditions. Suppose we would like a machine that takes a string of arbitrary characters and replaces every sequence of "A"s, however long, by a single "A". (This is an elementary form of compression.) For example, the string "BAAAAB" would become "BAB", and the string "AAAAAAAAABAA" would become "ABA." This can be done with a finite state machine having three states: State 1 is "I haven't seen any characters yet;" State 2 is "I just saw an "A";" State 3 is "I just saw something other than "A"." To specify a finite state machine, one must list the states and then define what state the machine is to transition into, given what state it is in and what input character it sees. Here the list of states is {State 1, State 2, State 3}. At the beginning of a string, the machine is in State 1. Being in State 1 (having not seen any characters yet), if the next character is also an "A", then the machine should output nothing and transition to State 2 (having just seen an "A"). Being in State 1, if the next character in the input string is something other than "A", the machine should output that character and transition to State 3 (just having seen a non-"A"). Being in State 2 (having just seen an "A"), if the machine sees an "A" it should output nothing and remain in State 2. (By outputting nothing but the first "A" in a sequence of "A"s, the machine achieves compression.) Being in State 2 (having just seen an "A"), if the machine sees anything other than "A", it should output that character and transition to State 3 (having just seen a non-"A"). Being in State 3 (having just seen a non-"A"), if the machine sees an "A" it should output an "A" and transition to State 2 (having just seen an "A"). Being in State 3 (having just seen a non-"A"), if the machine sees another non-"A", it should output that character and remain in State 3.

24. The set of transitions between states is called a "transition table," and may be expressed in various ways. The simplest is a "transition table." The transition table for this machine looks like:

	Character is an "A"	Character is not an "A"
State 1 (Start)	Output "A"; Transition to State 2.	Output the character; Transition to State 3.
State 2 (Just saw an "A")	Output NOTHING; Remain in State 2.	Output the character; Transition to State 3.
State 3 (Just saw a non-"A")	Output "A"; Transition to State 2.	Output the character; Remain in State 3.

Such a table is referred to as a state transition table for the finite state machine. Because compression algorithms transform strings into other strings that are (hopefully) shorter, it is apparent that finite state machines are important in compression. While all finite state machines possess at least an implied transition table, that table does need to be explicitly represented in tabular form. It may, instead, be implemented in software instructions or embodied in some other manner.

25. A "global state machine" is defined in the '651 Patent as follows: "a global state system is constructed based on a priori knowledge of the data stream model, e.g., the packet type frequency and structure of the broadcast model. By way of example, one model for financial data may comprise four global states representing: a beginning of packet, an options packet, a NYSE (New York Stock Exchange) packet and some other packet type." '651 Patent, 10:53-59. The idea of a "global" state machine is that its transitions depend only on the presumed characteristics of the data stream in general, and not on the actual characters that are present in the data stream. Once the identity of the data stream is known, e.g. that it is from the New York Stock Exchange, the transition table for that stream is fixed and does not vary based on the actual characters in the data stream.

26. By contrast, a "local" state machine has a transition table whose entries may vary in time based on the actual characters that appear in the data stream. This is explained in the specification: "In a preferred embodiment, the n-tuples comprise character sequences having 1, 2 and 3 characters. Using the acquired counts, sub-states (or 'local states') of the predefined global states are constructed based on previous characters in the data stream." '651 Patent at 11:2-6. Thus, in a "local" state machine, new states and transitions can be generated based on the actual characters in the data stream.

27. The principal substantive difference between the parties' constructions for "global state machines" is that Defendants require the machines to have "prior knowledge of at least the data stream packet structure." This is incorrect because (1) there is no such teaching or limitation in the patent; and (2) the global state machine need not have any "knowledge" of the packet structure. The '651 Patent states that "The 'context' in which a character (or character sequence) is encoded in a given broadcast stream is based on a 'global state' that represents packet type and large-scale structure and the previous few characters." '651 Patent at 10:47-50. It is the "large scale structure," by which is meant the properties of the data stream, on which the global state machine is based, not the "data stream packet structure," as Defendants contend.

28. Realtime's construction—"hardware, microinstruction code, or application program whose state transitions are based on prior knowledge of at least some aspects of the content or structure of the financial data stream"—captures the teaching of the specification.

D. CONCLUSIONS

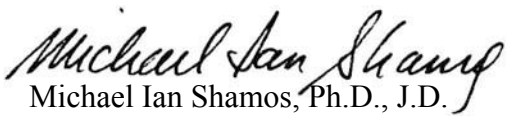
29. Realtime's constructions of the disputed claim terms are consistent with the specification and the understanding of one of ordinary skill in the art. Defendants' constructions

are either crafted without regard to the specification or attempt to import limitations of a disclosed embodiment into the claims.

E. SIGNATURE AND STATEMENT OF TRUTH

30. I confirm that the contents of this Declaration are true to the best of my knowledge and belief insofar as it states facts and that it contains my honest opinions on the matters upon which I have been asked to give them.

Dated: March 26, 2012



Michael Ian Shamos, Ph.D., J.D.